

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 2 年 9 月 9 日
Date of Application:

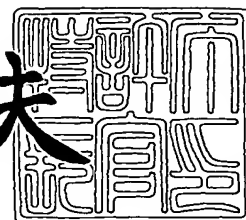
出 願 番 号 特 願 2 0 0 2 - 2 6 2 7 7 7
Application Number:
[ST. 10/C] : [J P 2 0 0 2 - 2 6 2 7 7 7]

出 願 人 キヤノン株式会社
Applicant(s):

2 0 0 3 年 1 0 月 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 4787012

【提出日】 平成14年 9月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 プリントシステム、情報処理装置、プリント方法、コンピュータプログラム、及びコンピュータ読み取り可能な記憶媒体

【請求項の数】 12

【発明者】

 【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会社
社内

 【氏名】 友松 美明

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100090273

 【弁理士】

 【氏名又は名称】 國分 孝悦

 【電話番号】 03-3590-8901

【手数料の表示】

 【予納台帳番号】 035493

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9705348

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プリントシステム、情報処理装置、プリント方法、コンピュータプログラム、及びコンピュータ読み取り可能な記憶媒体

【特許請求の範囲】

【請求項 1】 アプリケーションソフトウェア及びプリンタドライバがインストールされた情報処理装置と、画像出力装置とが接続されたプリントシステムであって、

前記アプリケーションソフトウェアが 1 ページ分のデータを上から下に順に出力する処理を実行し、前記プリンタドライバが 1 ページ分のデータをバンド状に分割した状態で前記画像出力装置に出力する処理を実行する並行処理モードを有することを特徴とするプリントシステム。

【請求項 2】 前記アプリケーションソフトウェアから前記並行処理モードが指定された場合に、前記プリンタドライバが前記並行処理モードとするか否かを判定することを特徴とする請求項 1 に記載のプリントシステム。

【請求項 3】 前記アプリケーションソフトウェアが前記並行処理モードを指定する際に前記アプリケーションソフトウェアと前記プリンタドライバのみが知る独自の方式を用いることを特徴とする請求項 2 に記載のプリントシステム。

【請求項 4】 前記並行処理モードに移行するに際して、前記プリンタドライバが、前記情報処理装置にインストールされている基本ソフトウェアのスプール設定をオフにすることを特徴とする請求項 1 ～ 3 のいずれか 1 項に記載のプリントシステム。

【請求項 5】 前記並行処理モードに移行するに際して、前記プリンタドライバが、前記情報処理装置にインストールされている基本ソフトウェアのバンディング処理を停止するよう前記基本ソフトウェアに対して通知することを特徴とする請求項 1 ～ 4 のいずれか 1 項に記載のプリントシステム。

【請求項 6】 前記プリンタドライバは、前記アプリケーションソフトウェアから出力されたデータと、前記画像出力装置の出力するバンドの位置との関係を判定し、その結果に応じてバンド状に分割したデータを前記画像出力装置に出力する処理を実行することを特徴とする請求項 1 ～ 5 のいずれか 1 項に記載のプ

リントシステム。

【請求項 7】 前記アプリケーションソフトウェアは、1 ページ分のデータを上から下に順に出力する際にバンド状に分割して出力することを特徴とする請求項 1～6 のいずれか 1 項に記載のプリントシステム。

【請求項 8】 アプリケーションソフトウェア及び画像出力装置を利用するためのプリンタドライバがインストールされた情報処理装置であって、

前記アプリケーションソフトウェアが 1 ページ分のデータを上から下に順に出力する処理を実行し、前記プリンタドライバが 1 ページ分のデータをバンド状に分割して前記画像出力装置に出力する処理を実行する並行処理モードを有することを特徴とする情報処理装置。

【請求項 9】 アプリケーションソフトウェア及び画像出力装置を利用するためのプリンタドライバがインストールされた情報処理装置を用いてプリントするプリント方法であって、

前記アプリケーションソフトウェアに 1 ページ分のデータを上から下に順に出力する処理を実行させ、前記プリンタドライバに 1 ページ分のデータをバンド状に分割して前記画像出力装置に出力する処理を実行させることを特徴とするプリント方法。

【請求項 10】 プリントのためのデータをコンピュータにインストールされたプリンタドライバに出力する処理を前記コンピュータに実行させるコンピュータプログラムであって、

前記プリンタドライバに並行処理モードを指定する処理と、

前記並行処理モードにおいて 1 ページ分のデータを上から下に順に出力する処理とを前記コンピュータに実行させることを特徴とするコンピュータプログラム。

【請求項 11】 コンピュータにインストールされたアプリケーションソフトウェアからプリントするデータを受け取って、前記コンピュータに接続された画像出力装置に出力する処理を前記コンピュータに実行させるコンピュータプログラムであって、

前記アプリケーションソフトウェアからの指定により並行処理モードとなった

場合に、1 ページ分のデータをバンド状に分割した状態で前記画像出力装置に出力する処理を前記コンピュータに実行させることを特徴とするコンピュータプログラム。

【請求項 1 2】 請求項 1 0 又は 1 1 に記載のコンピュータプログラムを格納したことを特徴とするコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、アプリケーションソフトウェア等で作成した文字、図形等の描画データをラスタイメージの画像に展開して画像出力装置に送信するプリントシステム、情報処理装置、プリント方法、コンピュータプログラム、及びコンピュータ読み取り可能な記憶媒体に関するものである。

【0 0 0 2】

【従来の技術】

近年のカラープリンタやスキャナの低価格化から、コンピュータにカラープリンタ、スキャナを接続して、安価で手軽にカラー原稿をコピーすること等が可能になった。

【0 0 0 3】

【発明が解決しようとする課題】

通常、コンピュータにカラープリンタ、スキャナを接続してコピーする場合、コンピュータ上のアプリケーションソフトウェアが、一旦スキャナから画像を読み取り、読み取った画像をカラープリンタに出力する方法が取られる。

【0 0 0 4】

一般的に普及しているコンピュータにMicrosoft社のWindows (R) を OS (Operating System) として使用する場合、OS の仕様上、アプリケーションソフトウェアから出力されたデータは、ページ分のデータが終了するまでは印刷が開始されない。

【0 0 0 5】

しかし、スキャナからの画像の取り込み、及び、画像データのプリンタへの印

字はそれぞれ時間がかかるため、ユーザからのコピー開始要請から印刷終了までに多くの時間がかかってしまうという問題があった。

【0006】

本発明は上記の点に鑑みてなされたものであり、アプリケーションソフトウェアがスキャナから読み込んだ画像をプリンタで印刷する等、アプリケーションソフトウェアがプリンタへの出力データを作成するのに時間がかかる場合には、プリント処理を並行して行うことでプリント終了までを高速に行うことを目的とする。

【0007】

【課題を解決するための手段】

本発明のプリントシステムは、アプリケーションソフトウェア及びプリンタドライバがインストールされた情報処理装置と、画像出力装置とが接続されたプリントシステムであって、前記アプリケーションソフトウェアが1ページ分のデータを上から下に順に出力する処理を実行し、前記プリンタドライバが1ページ分のデータをバンド状に分割した状態で前記画像出力装置に出力する処理を実行する並行処理モードを有する点に特徴を有する。

【0008】

本発明の情報処理装置は、アプリケーションソフトウェア及び画像出力装置を利用するためのプリンタドライバがインストールされた情報処理装置であって、前記アプリケーションソフトウェアが1ページ分のデータを上から下に順に出力する処理を実行し、前記プリンタドライバが1ページ分のデータをバンド状に分割して前記画像出力装置に出力する処理を実行する並行処理モードを有する点に特徴を有する。

【0009】

本発明のプリント方法は、アプリケーションソフトウェア及び画像出力装置を利用するためのプリンタドライバがインストールされた情報処理装置を用いてプリントするプリント方法であって、前記アプリケーションソフトウェアに1ページ分のデータを上から下に順に出力する処理を実行させ、前記プリンタドライバに1ページ分のデータをバンド状に分割して前記画像出力装置に出力する処理を

実行させる点に特徴を有する。

【 0 0 1 0 】

本発明のコンピュータプログラムは、プリントのためのデータをコンピュータにインストールされたプリンタドライバに出力する処理を前記コンピュータに実行させるコンピュータプログラムであって、前記プリンタドライバに並行処理モードを指定する処理と、前記並行処理モードにおいて 1 ページ分のデータを上から下に順に出力する処理とを前記コンピュータに実行させる点に特徴を有する。

【 0 0 1 1 】

また、本発明の他のコンピュータプログラムは、コンピュータにインストールされたアプリケーションソフトウェアからプリントするデータを受け取って、前記コンピュータに接続された画像出力装置に出力する処理を前記コンピュータに実行させるコンピュータプログラムであって、前記アプリケーションソフトウェアからの指定により並行処理モードとなった場合に、1 ページ分のデータをバンド状に分割した状態で前記画像出力装置に出力する処理を前記コンピュータに実行させる点に特徴を有する。

【 0 0 1 2 】

本発明のコンピュータ読み取り可能な記憶媒体は、前記本発明のコンピュータプログラムを格納した点に特徴を有する。

【 0 0 1 3 】

【発明の実施の形態】

以下、図面を参照して、本発明のプリントシステム、情報処理装置、プリント方法、コンピュータプログラム、及びコンピュータ読み取り可能な記憶媒体の実施の形態について説明する。

【 0 0 1 4 】

図 1 には、本実施の形態におけるプリントシステムの機能構成の概略を示す。例えば一般的に普及しているパーソナルコンピュータ 1 0 1 に Microsoft 社の Windows (R) を OS (Operating System: 基本ソフトウェア) として使用し、任意の印刷可能なアプリケーションソフトウェア 1 0 2 (以下、アプリケーション 1 0 2) をインストールし、プリンタ 1 0 5 及びスキャナ 1 0 7 を接続した形態が考

えられる。

【0 0 1 5】

アプリケーション 1 0 2 は、OS 1 0 3 を経由し、スキャナドライバ 1 0 6 を使ってスキャナ 1 0 7 から画像データを入力し、プリンタドライバ 1 0 4 を使ってプリンタ 1 0 5 に画像を出力する。

【0 0 1 6】

通常、スキャナ 1 0 7 からの入力、及び、プリンタ 1 0 5 での出力には時間がかかる。そこで、スキャナ 1 0 7 からデータを入力しながらプリンタ 1 0 5 で印刷するという並行処理を行うことにより高速にコピーすることが可能となる。

【0 0 1 7】

ところで、アプリケーション 1 0 2 がプリンタ 1 0 5 に画像データを出力しても、

1 - 1) OS のスプーリング処理

1 - 2) Windows (R) アプリケーション・プログラミング・インターフェイス (API) の影響

により、1 ページ分のデータが作成されるまで OS 1 0 3 からプリンタドライバ 1 0 4 にデータが出力されず、1 ページ分のスキャン後にプリントしてしまうという問題があった。したがって、並行処理を可能にするには、これらの点についてなんらかの対策をたてる必要がある。

【0 0 1 8】

1 - 1) OS のスプーリング処理

アプリケーション 1 0 2 からの印刷時、印刷が終了する前にアプリケーション 1 0 2 が次の処理を開始できるように、OS 1 0 3 にはバックグラウンド印刷機能がある。この機能は、OS 1 0 3 がアプリケーション 1 0 2 からの出力コマンドを一旦 EMF (Enhanced Meta File) データとしてスプールし、アプリケーション 1 0 2 が次の処理を開始できるようにするものである。アプリケーション 1 0 2 が次の作業を行っているときに、OS 1 0 3 がスプールされた EMF データをプリンタドライバ 1 0 4 に出力し印刷を行う。

【0 0 1 9】

EMF スプール設定はプリンタドライバ 1 0 4 毎に設定可能で、ユーザが切り替えたり、プリンタドライバ 1 0 4 で禁止したりすることも可能であるが、通常、プリンタドライバ 1 0 4 がインストールされた時点で EMF スプールは ON の設定になっており、そのまま使用されている。

【 0 0 2 0 】

特に、OS 1 0 3 がこの EMF データによるスプールファイルを利用して、最後のページから印刷する逆順印刷や、1 枚の用紙に複数ページを印刷するという N - u p 印刷等多くの機能を実現していることから、バックグラウンド印刷機能が不要だからと、ユーザやプリンタドライバ 1 0 4 がこの EMF スプール設定を OFF にすると、逆順印刷や N - u p 印刷ができなくなるという問題もある。

【 0 0 2 1 】

そこで、スキャナ 1 0 7 からのコピー並行処理等の並行処理を行う場合のみ、プリンタドライバ 1 0 4 で EMF スプール設定を OFF にすることで、ユーザが意識することなく並行処理できるようにする。また、逆順印刷や N - u p 印刷等の EMF スプール機能を利用する機能をユーザが ON にしている場合には、並行処理を行わないという処理も可能となる。

【 0 0 2 2 】

1 - 2) Windows (R) アプリケーション・プログラミング・インターフェイス (API) の影響

Windows (R) API の仕様ではアプリケーション 1 0 2 はページ内に任意の順で出力できるため、ページ終了命令まで印刷処理を開始できない。例えば、図 2 に示すデータを印刷する場合、アプリケーション 1 0 2 は以下の順番で処理を行う。

1. ページ開始命令
2. 背景のビットマップ 2 0 1 を出力 (図 2 では現れていないが、色の薄い模様が描かれている)
3. 下方のイラスト 2 0 4 を出力
4. 文字列『あけましておめでとうございます』 2 0 2 を出力
5. 文字列『去年は…元旦』 2 0 3 を出力

6. ページ終了命令

【0023】

このように、一旦描画した背景の上に文字等を再度描画することが可能であるため、ページ終了命令までページすべての位置の印字データが決まらず印刷を開始できない。

【0024】

Windows (R) のプリントシステムにおいては、このため以下の方法で印刷を行っている。

【0025】

2-1) バンディングドライバの場合

アプリケーション102等で作成した文字、図形等の画像データをラスティメージで画像出力装置に出力するタイプのプリンタ105の場合、一旦ホストコンピュータであるパーソナルコンピュータ101上の仮想的なページメモリに展開後、画像出力装置に出力している。

【0026】

この際、ホストコンピュータのメモリ効率化等の関係で、この仮想ページメモリを幾つかのバンドに分割処理し、バンド毎にOS103からの画像データ等の出力要求に従いバンドメモリにイメージで描画し、そのバンド領域すべての描画が終了したというOS103からのコントロールに従いバンドメモリに展開されたデータを画像出力装置に出力している。

【0027】

図3はバンド分割の概要を示す図であるが、OS103は各バンドに属す出力コマンドを抽出し、プリンタドライバ104に出力する。そのため、OS103は、一旦ページ分の描画コマンドを保持し、バンド毎にプリンタドライバ104に出力要求する、例えば、バンド1であればバンド1に属す、

1. 背景ビットマップ201 (図3では現れていないが、色の薄い模様が描かれている)
2. 文字列『あけましておめでとうございます』202
3. 文字列『去年は…元旦』203

をプリンタドライバ104に出力要求する。

【0028】

要求を受けたプリンタドライバ104は、上記出力コマンドのうちバンド1に属す領域だけをプリンタ105に出力し、OS103に処理終了と次のバンド位置を通知する。終了通知を受けたOS103は、以降のバンド2、3、…を順にバンド1同様に対象とするバンドに属すデータだけをプリンタドライバ104に出力する。

【0029】

以上述べたようにバンディングドライバの場合には、OS103内でページ分のデータを一旦保持しプリンタドライバ104に出力している。

【0030】

2-2) アンバンディングドライバの場合

プリンタ105がページメモリや1ページ分の描画データ記憶手段を持ちページ分の描画データをプリンタ105内部で保持できる場合や、プリンタドライバ104内にスプール機能を持ちプリンタドライバ104内部でページ分の描画データを保持できる場合には、プリンタドライバ104はOS103に対しアンバンディングタイプで処理できることを通知する。

【0031】

アンバンディングのプリンタドライバ104の場合、OS103はEMFスプール設定がされていなければ、アプリケーション102からの出力を即座にプリンタドライバ104及びプリンタ105に出力する。出力コマンドを受けたプリンタドライバ104及びプリンタ105は、Windows(R) APIの関係でページ終了命令があるまでデータが確定しないため、内部に出力データを保持し、ページ終了命令があった時点で印刷処理を開始する。

【0032】

このように、Windows(R) APIを満たす、すべてのアプリケーションからの印刷を満たそうとすると、ページ終了までは印刷できないことになる。

【0033】

そこで、プリンタドライバ104に並行処理のモードを作成し、

- ・アプリケーション 1 0 2 は常にプリンタ 1 0 5 の用紙搬送方向（用紙の上から下）、すなわち 1 ページ分の上から下にデータを順に出力する。

- ・並行処理モードは上記のルールを知るアプリケーション 1 0 2 からのみ指定できるようにする。

- ・プリンタドライバ 1 0 4 は OS 1 0 3 に対しアンバンディングであると宣言し、バンディング処理はプリンタドライバ 1 0 4 で行うことで、並行処理できるようにする。

【 0 0 3 4 】

図 4 は、スキャナ 1 0 7 でのスキャン(Scanで表示)時間とプリンタ 1 0 5 でのプリント(Print及びPで表示)時間を示したものである。通常は、スキャナ 1 0 7 でのスキャン後、OS 1 0 3 がスプールされたEMF スプールファイルの印刷をプリンタ 1 0 5 に要求し、プリントが行われるのに対して、並行処理時は、スキャンしたデータを次の領域のスキャン中にプリントするので、コピー開始から印刷終了までの時間は短くなる。

【 0 0 3 5 】

以下、図 6 ～ 9 を参照して本実施の形態における処理の流れを詳細に説明する。図 6 はアプリケーション 1 0 2 の処理、図 7 はプリンタドライバ 1 0 4 の初期化処理、図 8 はプリンタドライバ 1 0 4 の印刷処理、図 9 はプリンタドライバ 1 0 4 のページ終了処理の流れを示すフローチャートである。

【 0 0 3 6 】

まず、アプリケーション 1 0 2 はプリンタドライバ 1 0 4 に対し並行処理モードとするように指示する（ステップ S 1 0 1）。上述したように並行処理モードでは特定のデータ出力順序のみでしか使えないため、アプリケーション 1 0 2 はこの出力順序で印刷する場合のみプリンタドライバ 1 0 4 に並行処理モードでの印刷を指定する。ユーザやこの出力順序等を知らないアプリケーションが並行処理モードでの印刷を指定し、アプリケーション 1 0 2 が既に印刷済みの領域に対し印刷を要求すると正常に印刷できなくなるためである。例えば、図 2 に示すデータの場合には、背景ビットマップ 2 0 1 をアプリケーション 1 0 2 が出力した時点でページ全体の印刷を行ってしまい、その後の文字等が印刷されなくなって

しまう。そのため、Windows (R) の仕様にはない独自の設定方法が好ましい。例えば、プリンタドライバ 1 0 4 に独自の関数を設け、並行処理に対応したアプリケーション 1 0 2 がその関数を使って設定する。

【 0 0 3 7 】

初期化要求されたプリンタドライバ 1 0 4 の処理を図 7 を参照して説明する。まず、プリンタドライバ 1 0 4 はアプリケーション 1 0 2 から並行処理モードが指定されたかどうかを判定する（ステップ S 2 0 1）。

【 0 0 3 8 】

ステップ S 2 0 1 で Y e s と判定された場合には、ドライバ設定が並行処理に適するかどうかを判定する。並行処理モードでは、E M F スプールを禁止し、バンディング処理をプリンタドライバ 1 0 4 内部で行う。しかし、O S 1 0 3 は E M F スプールファイルを使って多くの機能を実現している。例えば、アプリケーション 1 0 2 からの複数ページが印刷される際、全てのページを E M F スプールファイルとしてスプールし、最後のページから順番に出力するという逆順印刷等の機能は使えなくなる。同じくバンディング処理を利用することで実現できる機能も使えなくなる。

【 0 0 3 9 】

そこで、これらの機能を使って印刷するようにユーザが指示した場合には、並行処理モードを行わずユーザが設定した機能をつけて印刷を行うため、ここでは現在の設定が E M F スプール又はバンディングが必要かを判定する（ステップ S 2 0 2）。ただし、機能によっては並行処理を優先して、その機能を使わずに印刷することも可能である。

【 0 0 4 0 】

ステップ S 2 0 2 で N o と判定された場合には、並行処理を行うために、まず対象とするジョブの E M F 処理を禁止する（ステップ S 2 0 3）。禁止は、O S 1 0 3 からジョブ（アプリケーションの印刷単位）単位に初期化される際に設定し O S 1 0 3 に返す DEVINFO というデータに、GCAPS_DONTJOURNAL というフラグをセットすることで実現できる。このフラグは初期化するジョブに対してのみ有効であり、それ以降の印刷には影響しない。

【 0 0 4 1 】

続いて、OS 1 0 3 に対しアンバンディングでの処理であると宣言する（ステップ S 2 0 4）。プリンタドライバ 1 0 4 は Surface と呼ばれる仮想デバイスであるデータを作成し、初期化の際に OS 1 0 3 に返すが、OS 1 0 3 内の EngMarkBandingSurface. という関数をコールすることで、作成した Surface をバンディングドライバ用に設定することができる。OS 1 0 3 は、バンディング用の Surface を返されるとバンディングドライバとして処理する。一方、EngMarkBandingSurface. をコールせずに返すとアンバンディングとして処理する。ここでは OS 1 0 3 がアンバンディングとして処理する必要があるので、EngMarkBandingSurface. をコールせずに返す。この設定も対象とするジョブに対してのみ有効であり、それ以降の印刷には有効とならない。

【 0 0 4 2 】

一方、ステップ S 2 0 1 で No 又はステップ S 2 0 2 で Yes だった場合、つまりアプリケーション 1 0 2 から並行処理モードの指示がされなかったか、並行処理を指定されたが EMF スプール又はバンディングが必要な設定であるとプリンタドライバ 1 0 4 が判断し並行処理を止めた場合には、ステップ S 2 0 3、S 2 0 4 とは逆に、まず OS 1 0 3 からジョブ単位の初期化時に GCAPS_DONTJOURNAL フラグをセットせずに DEVINFO を設定し（ステップ S 2 0 5）、Surface を EngMarkBandingSurface. にコールしてから OS 1 0 3 に返す（ステップ S 2 0 6）。

【 0 0 4 3 】

図 6 に説明を戻して、プリンタドライバ 1 0 4 の初期化が終わったら、アプリケーション 1 0 2 はスキャナ 1 0 7 から入力した画像をプリンタ 1 0 5 に出力するが、スキャナ 1 0 7 から入力した画像は原稿の 1 ページをバンド状に分割し、そのバンド単位で入出力を繰り返す。すなわち、アプリケーション 1 0 2 はスキャナドライバ 1 0 6 経由でスキャナ 1 0 7 から 1 バンド分の画像を入力し（ステップ S 1 0 2）、入力した 1 バンド分の画像をプリンタドライバ 1 0 4 経由でプリンタ 1 0 5 に出力する（ステップ S 1 0 3）。その結果、スキャナ 1 0 7 から次のバンドの画像を入力しているときに、プリンタ 1 0 5 が前のバンドの印刷処理を行い並行で処理できるようになる。このバンドのサイズはプリンタドライバ 1 0 4 の

バンドサイズと同じでもよいが、異なってもよい。

【 0 0 4 4 】

出力を要求されたプリンタドライバ 1 0 4 の処理を図 8 を参照して説明する。まず、出力要求されたイメージが対象としているプリンタドライバ 1 0 4 のバンド内に含まれるかを判定する。プリンタドライバ 1 0 4 のバンド位置は、例えば、最初は図 3 のバンド 1 のようにページの一番上に位置する。その後、バンド内のデータを全て印刷するに従い、バンド 2、バンド 3 というようにバンド位置を下方向に変えていく。ここでの判定は、出力要求された出力位置の一部でもバンドメモリ内に含まれるかどうかで判定する（ステップ S 3 0 1）。

【 0 0 4 5 】

バンド内に含まれると判定した場合、出力イメージのうち含まれている部分をバンドメモリに描画、つまり、出力イメージを印字領域に合わせて拡大縮小しバンドメモリにコピーする（ステップ S 3 0 2）。

【 0 0 4 6 】

ステップ S 3 0 1 で N o だった場合又はステップ S 3 0 2 の処理が終了したら、次に出力イメージとプリンタドライバ 1 0 4 のバンド位置との関係を判定する。図 5 は出力用紙上のプリンタドライバ 1 0 4 のプリンタバンド位置と出力イメージの関係を示す図であるが、同図（a）は出力イメージがプリンタバンド位置内に納まっているケース、（b）は出力イメージが次のバンドにもまたがっているケース、（c）は出力イメージがバンド位置より下のケースである。バンドは上から下に順次処理されているとする。

【 0 0 4 7 】

まず、出力イメージがバンドの最後まで描画したかを判定する。図 5 の例では、（b）のようにバンドの下端にも出力イメージが描画される場合には Y e s と判定され、（a）及び（c）の場合には N o と判定される（ステップ S 3 0 3）。

【 0 0 4 8 】

ステップ S 3 0 3 で N o だった場合、次に出力イメージが以降のバンドにも含まれるかを判定する。ここでは、（c）の場合にはバンドの下端よりも出力イメ

ージの方が下にあるのでYesと判定され、(a)の場合にはバンドの下端より下には出力イメージがないのでNoと判定される(ステップS304)。

【0049】

(a)のようにステップS304で以降のバンドには含まれないと判定された場合、出力要請されたイメージはバンドメモリへの描画は終了したが、現在のバンドメモリについては以降の描画命令で出力される可能性があるのでプリンタ105への出力は行わない、これで、ステップS103に対するプリンタドライバ104の処理は終了する。

【0050】

一方、(b)のようにステップS303でバンドの最後まで描画したと判定した場合、及び、(c)のようにステップS304で以降のバンドにも含まれると判定した場合、並行処理モードではアプリケーション102からのイメージ描画命令は上から下にしか来ないため、バンドの下端まで描画、又は、以降のバンドへの描画命令があったということは、その後このバンドへのイメージの出力要請はないと判定できる。そこで、バンドメモリに描画したイメージをプリンタコマンドに変換してプリンタ105に出力する(ステップS305)。

【0051】

次に、ステップS305で出力したバンドがページの最後のバンドであるかを判定する(ステップS306)。もし、ページの最後のバンドであると判定した場合には、ページ内のすべてのデータの出力が終了したことになるので、ここでもステップS103に対するプリンタドライバ104の処理は終了する。

【0052】

一方、ステップS306でNo、つまりすべてのバンドがまだ終了していないと判定されると、バンドの位置を次の位置に更新し(ステップS307)、ステップS301からの処理を繰り返す。

【0053】

図6に説明を戻して、プリンタドライバ104に出力後のアプリケーション102の処理について説明する。スキャナドライバ106からの画像を入力し、プリンタドライバ104にイメージを出力するという処理を繰り返し1ページ分行

う必要があるため、プリンタドライバ104の出力処理が終わったら、アプリケーション102はページ分のイメージをスキャンしたかを判定する（ステップS104）。

【0054】

ステップS104でNo、つまりページ分のスキャンが終了していないと判定した場合にはステップS102からの処理を繰り返す。一方、ステップS104でYes、つまりページ分のスキャンが終了したとすると、アプリケーション102は、プリンタドライバ104に対し、ページ終了命令を発行しアプリケーション102は処理を終了する。

【0055】

ページ終了命令を受けたプリンタドライバ104の処理を図9を参照して説明する。バンドメモリに描画され、まだプリンタ105に出力していないイメージをプリンタドライバ104はプリンタ105に出力する必要がある。まず、現在のバンドメモリに描画されているイメージがプリンタ105に出力されたものであるかを判定する。これはアプリケーション102から最後に出力されたイメージが、図5（b）、（c）のようにバンドの最後の位置まで描画し、ステップS305で既にプリンタ105に出力済みの場合があるからである（ステップS401）。

【0056】

ステップS401でYesと判定した場合には、ここでプリンタドライバ104の処理を終了する。一方、Noと判定した場合には、ステップS305同様バンドメモリに描画したイメージプリンタコマンドに変換してプリンタ105に出力する。これで、本処理を終了する。

【0057】

以上述べた実施の形態では、スキャナ107とプリンタ105を用いたコピーを例として説明したが、他のデバイスやネットワーク等から入力したデータのプリントや、アプリケーションが複雑な処理を行うため印字データ（イメージ以外の文字等も含む）を作成するのに時間がかかる場合にも同様の効果が得られることは言うまでもない。

【 0 0 5 8 】

(その他の実施の形態)

上述した実施の形態の機能を実現するべく各種のデバイスを動作させるように、該各種デバイスと接続された装置或いはシステム内のコンピュータに対し、上記実施の形態の機能を実現するためのソフトウェアのプログラムコードを供給し、そのシステム或いは装置のコンピュータ（C P U 或いは M P U）に格納されたプログラムに従って上記各種デバイスを動作させることによって実施したものも、本発明の範疇に含まれる。

【 0 0 5 9 】

また、この場合、上記ソフトウェアのプログラムコード自体が上述した実施の形態の機能を実現することになり、そのプログラムコード自体は本発明を構成する。そのプログラムコードの伝送媒体としては、プログラム情報を搬送波として伝搬させて供給するためのコンピュータネットワーク（L A N、インターネット等の W A N、無線通信ネットワーク等）システムにおける通信媒体（光ファイバ等の有線回線や無線回線等）を用いることができる。

【 0 0 6 0 】

さらに、上記プログラムコードをコンピュータに供給するための手段、例えばかかるプログラムコードを格納した記録媒体は本発明を構成する。かかるプログラムコードを記憶する記録媒体としては、例えばフレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、C D - R O M、磁気テープ、不揮発性のメモリカード、R O M等を用いることができる。

【 0 0 6 1 】

また、コンピュータが供給されたプログラムコードを実行することにより、上述の実施の形態の機能が実現されるだけでなく、そのプログラムコードがコンピュータにおいて稼働している O S（オペレーティングシステム）或いは他のアプリケーションソフト等と共同して上述の実施の形態の機能が実現される場合にもかかるプログラムコードは本発明の実施の形態に含まれることはいうまでもない。

【 0 0 6 2 】

さらに、供給されたプログラムコードがコンピュータの機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに格納された後、そのプログラムコードの指示に基づいてその機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部又は全部を行い、その処理によって上述した実施の形態の機能が実現される場合にも本発明に含まれることはいうまでもない。

【0063】

なお、上記実施の形態において示した各部の形状及び構造は、何れも本発明を実施するにあたっての具体化のほんの一例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。すなわち、本発明はその精神、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。

【0064】

【発明の効果】

以上述べたように本発明によれば、アプリケーションソフトウェアがスキャナから読み込んだ画像をプリンタで印刷する等、アプリケーションソフトウェアがプリンタへの出力データを作成するのに時間がかかる場合に、プリント処理を並行して行うことでプリント終了までを高速に行うことができる。

【図面の簡単な説明】

【図1】

実施の形態のプリントシステムの機能構成を示すブロック図である。

【図2】

アプリケーション102からの印刷データの一例を説明するための図である。

【図3】

印刷データのバンド分割の一例を説明するための図である。

【図4】

スキャナ107でのスキャン時間とプリンタ105でのプリント時間の関係を示す図である。

【図5】

出力用紙上のプリンタドライバ104のプリンタバンド位置と出力イメージの

関係を示す図である。

【図 6】

アプリケーション 1 0 2 の処理の流れを示すフローチャートである。

【図 7】

プリンタドライバ 1 0 4 の初期化処理の流れを示すフローチャートである。

【図 8】

プリンタドライバ 1 0 4 の印刷処理の流れを示すフローチャートである。

【図 9】

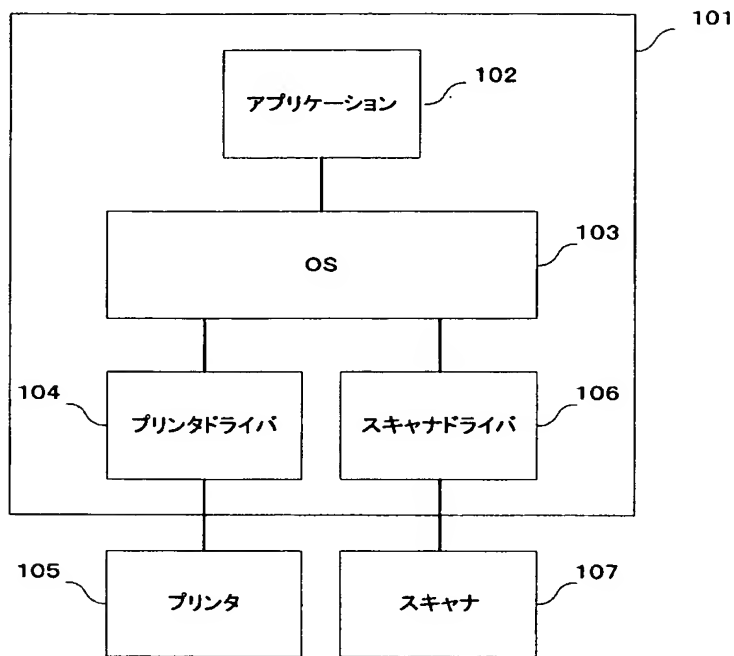
プリンタドライバ 1 0 4 のページ終了処理の流れを示すフローチャートである。
。

【符号の説明】

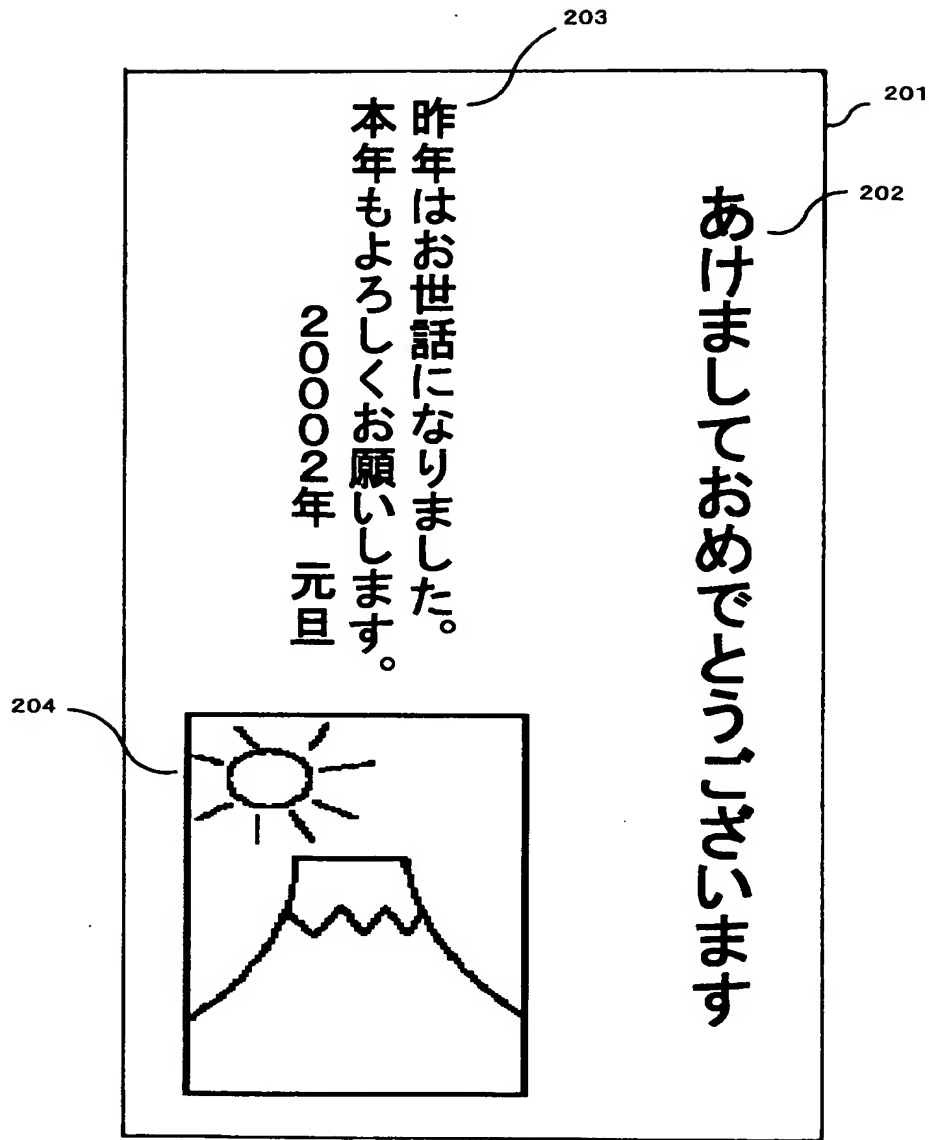
- 1 0 1 （情報処理装置である） パーソナルコンピュータ
- 1 0 2 アプリケーションソフトウェア
- 1 0 3 OS
- 1 0 4 プリンタドライバ
- 1 0 5 （画像出力装置である） プリンタ
- 1 0 6 スキャナドライバ
- 1 0 7 スキャナ

【書類名】 図面

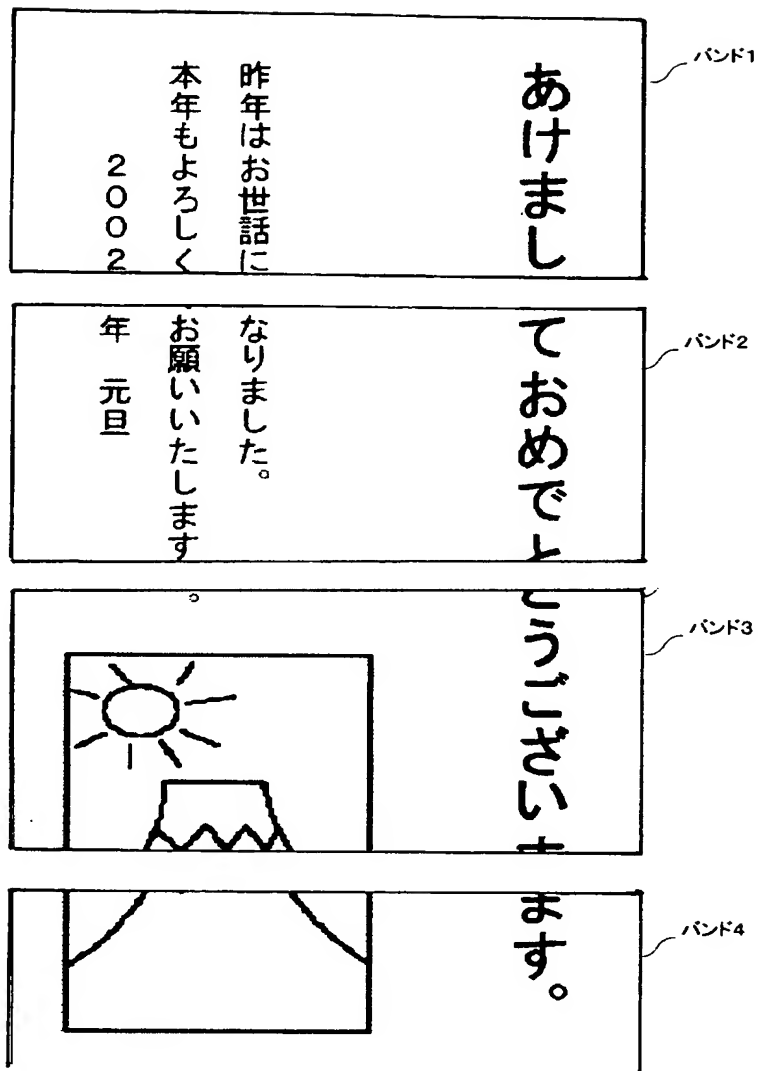
【図 1】



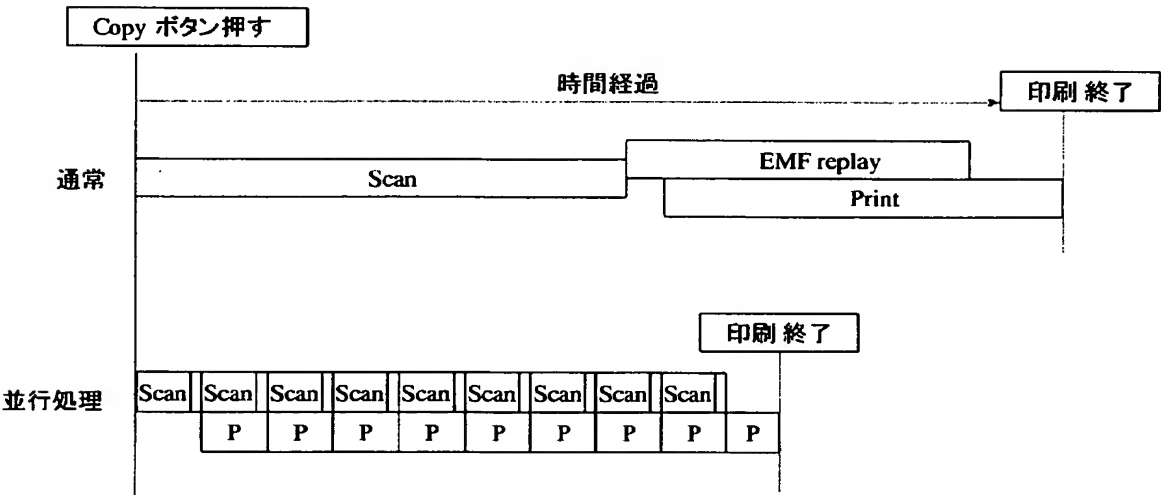
【図 2】



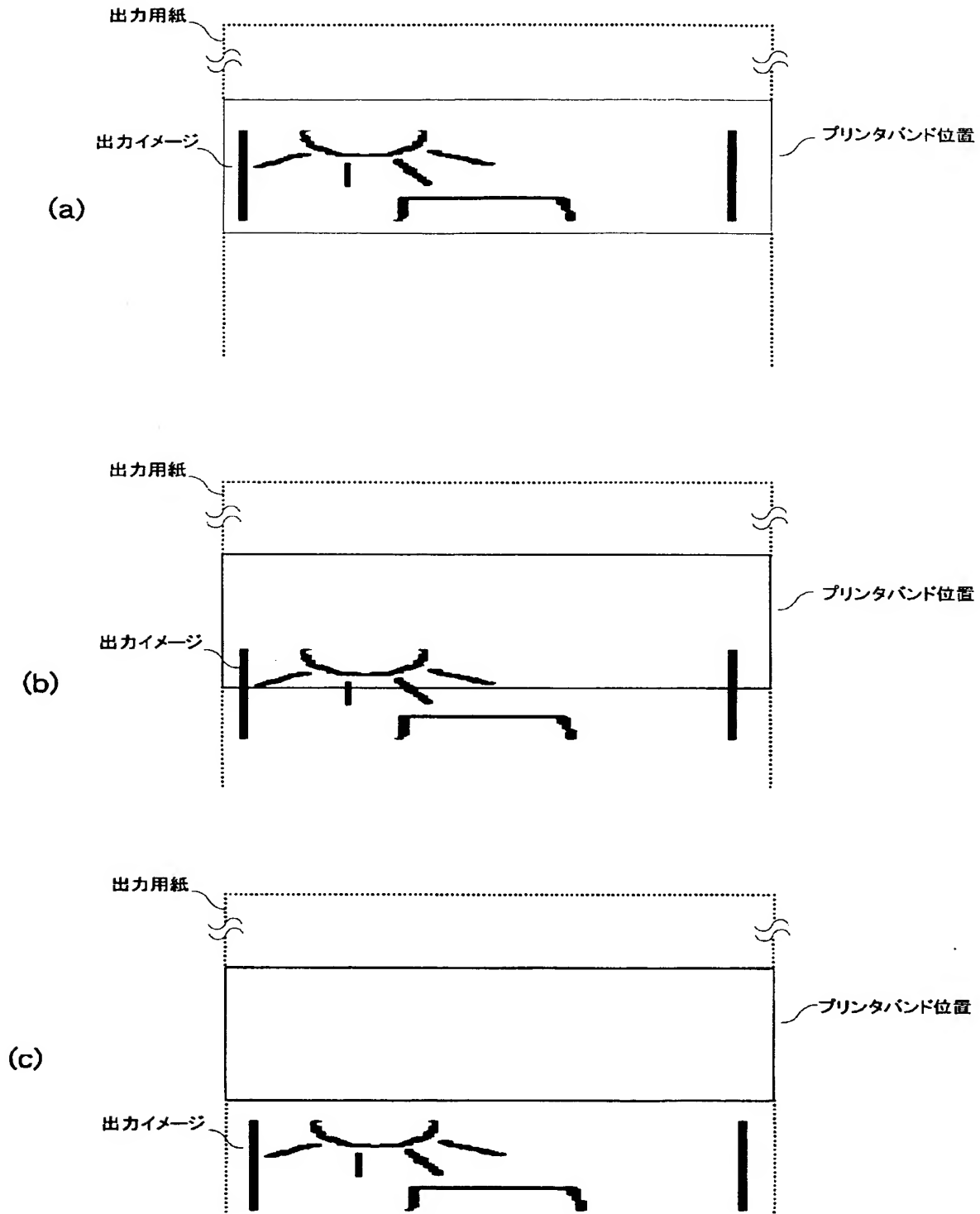
【図 3】



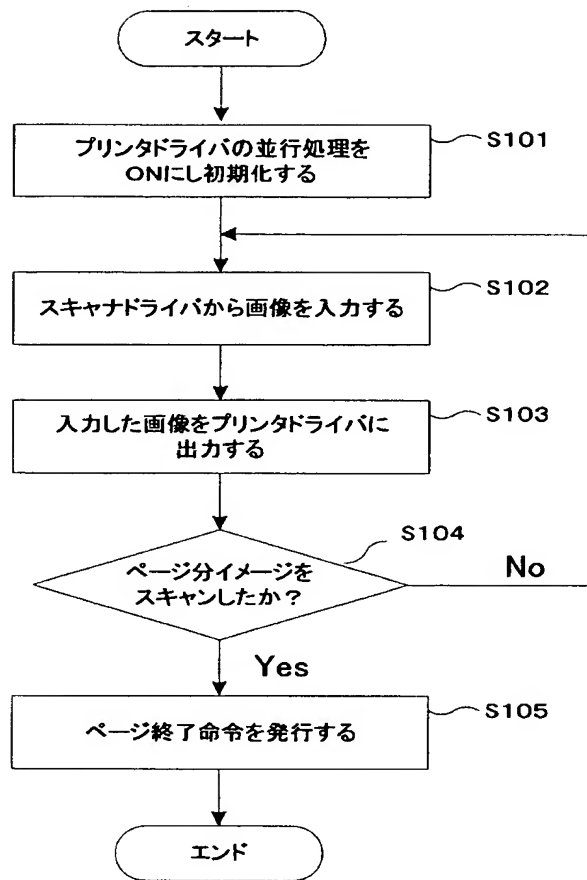
【図 4】



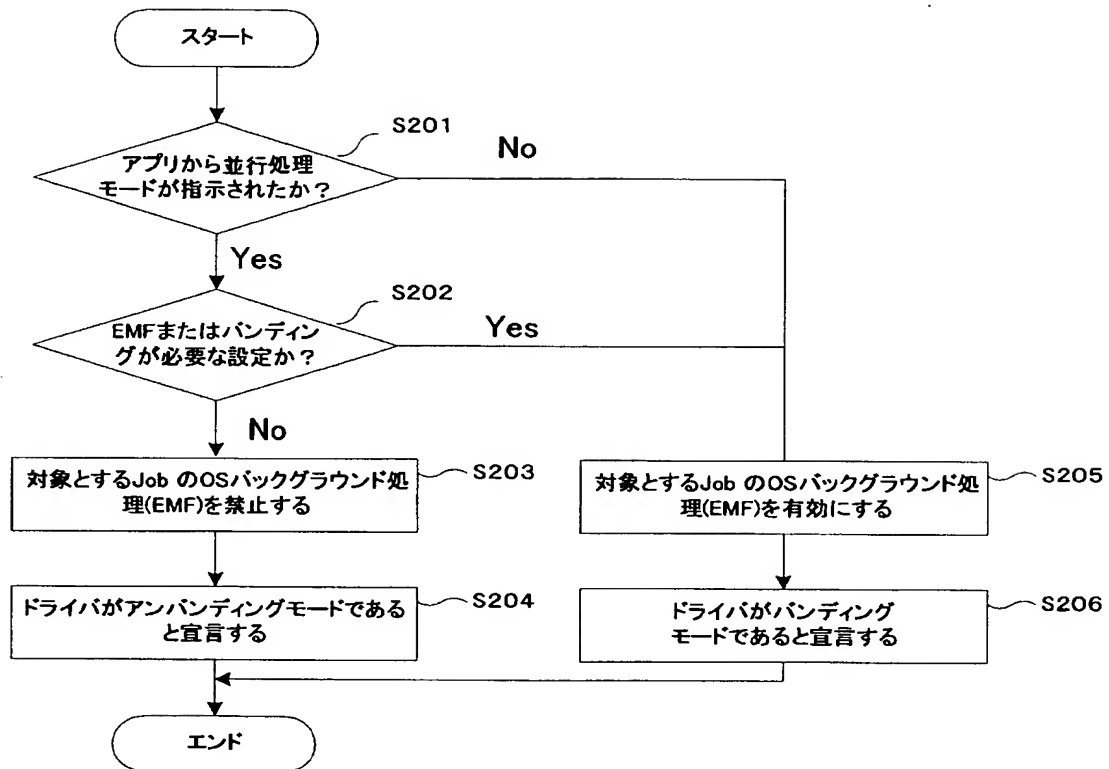
【図 5】



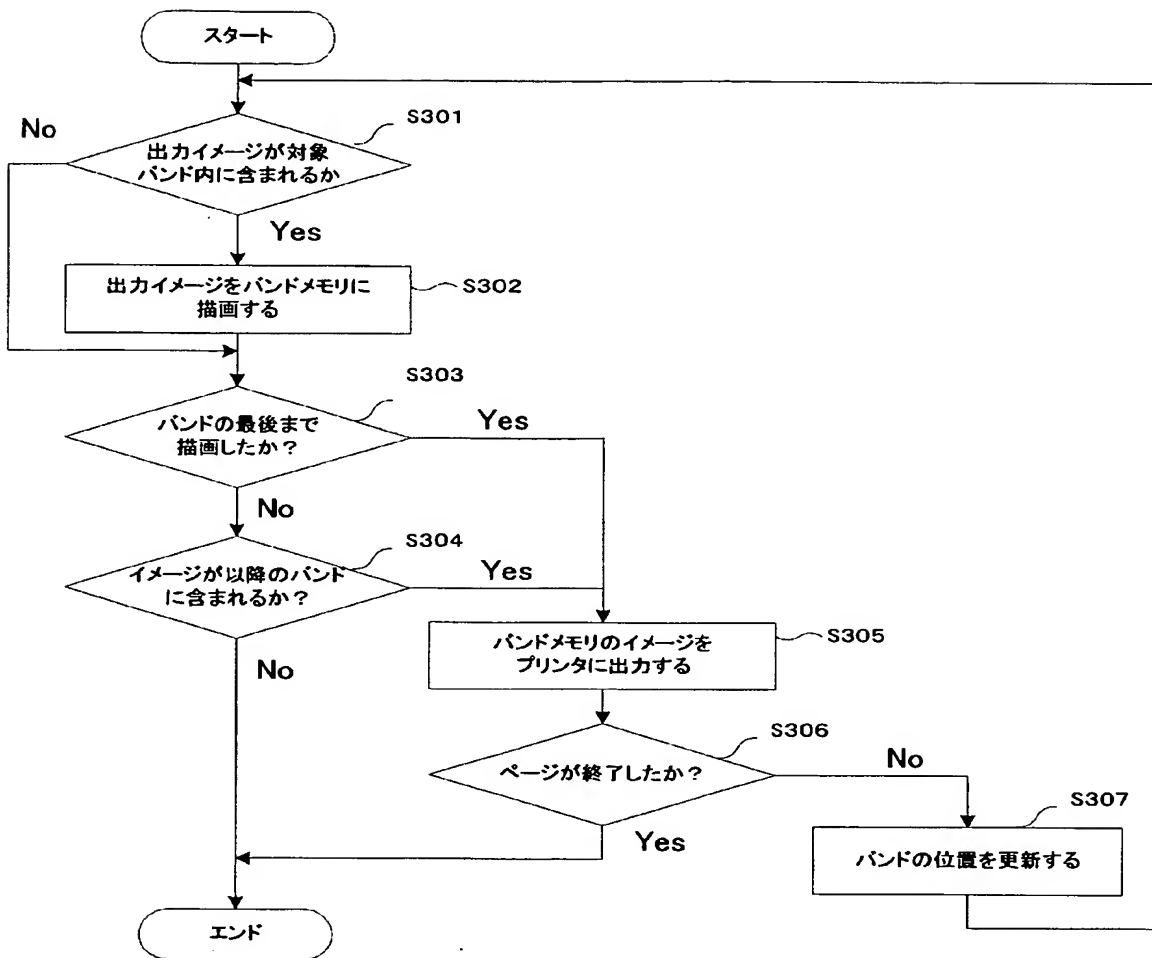
【図 6】



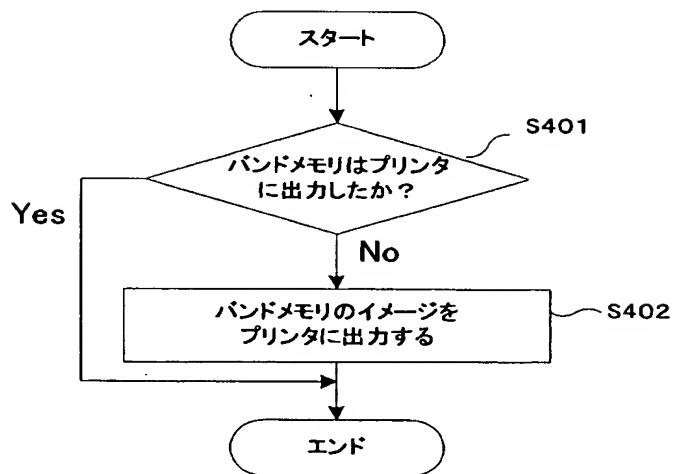
【図 7】



【図 8】



【図 9】



【書類名】 要約書

【要約】

【課題】 アプリケーションソフトウェアがプリンタへの出力データを作成するのに時間がかかる場合には、プリント処理を並行して行うことでプリント終了までを高速に行う。

【解決手段】 並行処理モードは所定のアプリケーション 1 0 2 からのみ指定できるようにしておき、並行処理モードでは、アプリケーション 1 0 2 は常にプリンタ 1 0 5 の用紙搬送方向（用紙の上から下）、すなわち 1 ページ分の上から下にデータを順に出力するとともに、プリンタドライバ 1 0 4 は O S 1 0 3 に対しアンバンディングであると宣言し、バンディング処理はプリンタドライバ 1 0 4 で行う。かかる並行処理時は、スキャンしたデータを次の領域のスキャン中にプリントするので、コピー開始から印刷終了までの時間は短くなる。

【選択図】 図 1

特願 2 0 0 2 - 2 6 2 7 7 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社